IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Learning Adaptive Controller for Hydraulic Machinery Automation

Fang Nan[1] and Marco Hutter[1]

***Abstract*—The automation of hydraulic machinery has the potential to improve productivity and reduce human labor in many industries. However, the complex dynamics of hydraulic actuators, variability from machine to machine, and system degradation over time make it challenging to design controllers for hydraulic machine automation. Consequently, existing approaches rely on manual tuning and data collection. In this paper, we propose an approach to train an adaptive controller for this problem. The controller can be trained purely in simulation, and at the time of deployment, it can adapt to the dynamics of the real system within minutes. After the adaptation, precise motion control can be achieved. We validated the approach by testing a single controller trained with the proposed method on two hydraulic machines that are distinctly different in size, application, and age. The results show comparable control performance of our general approach compared to previous methods, which rely on machine-specific data and training.**

***Index Terms*—Robotics and Automation in Construction; Hydraulic/Pneumatic Actuators; Reinforcement Learning**

I. Introduction

HYDRAULIC machinery is widely used in construction, agriculture, forestry, and disaster relief. In these tasks, they are often deployed in remote, risky, and harsh environments that are not safe for human operators. In addition, efficient operations of heavy machinery require skilled operators, which are not always easy to find. Autonomous heavy machinery is thus believed to be a promising direction, and it has been projected that robotic machinery will play a significant role in many industries in the near future [1], [2], [3].

In the automation of heavy machinery, a major challenge is the control design for the hydraulic actuators. Unlike electric motors, hydraulic cylinders are difficult to model and control due to the underlying cylinder topology and fluid dynamics. In most hydraulic machines, a multi-stage hydraulic circuit is used to control the cylinder as the force to move the main stage piston is too high to be directly controlled. Thus, the use of the pilot stage introduces additional nonlinearity and time delay to the actuation. The use of overlapping valves is common in hydraulic machinery to prevent leakage and increase efficiency, but it also brings additional nonlinearity and dead zones to the hydraulic dynamics.

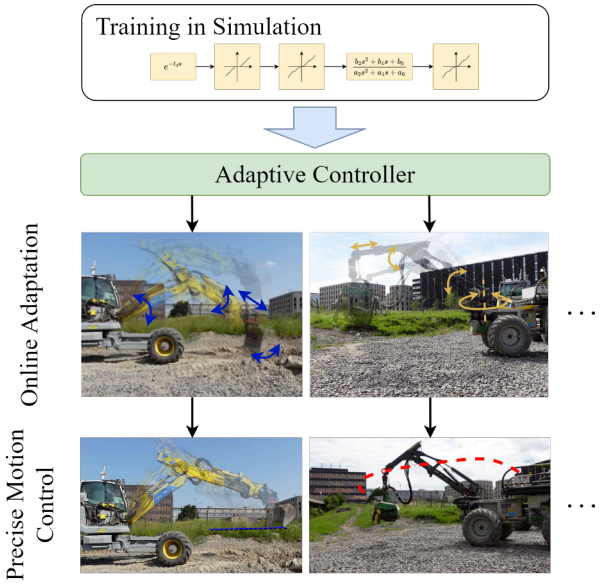


Fig. 1. The proposed method involves training a universal controller for hydraulic cylinders in a simulation environment of simplified hydraulic actuator models. The controller is then deployed on different machines. Online adaptation of cylinder dynamics can be carried out on each cylinder. After a short adaptation phase, the controller can be used for precise motion control. The pictures show two machines used in the experiments: a walking excavator (left) and a tree harvester (right).

Experienced machine operators can handle the complex dynamics of the hydraulic cylinders and perform coordinated control of multiple actuators precisely and efficiently. Human operators can also operate different machines and quickly adapt their skills to the new machine before tasks can be carried out effectively. Existing machine automation solutions, however, cannot achieve the same level of generalizability as they are often designed to be machine-specific. Some work uses controllers composed of a feedforward and a feedback module, both tuned by human experts [4], [5]. Data-driven modeling using machine learning approaches has also been used to construct models of the hydraulic systems, which are then used for offline controller training [6], [7], [8], [9]. These methods have been shown effective in controlling the hydraulic actuators in real-world experiments and achieved decent accuracy in tasks like end-effector trajectory tracking with the hydraulic arms on excavators. However, they heavily depend on machine-specific data for tuning, look-up table generation, or model training, and do not perform well in transferring to different machines without repeating these tedious processes.

Manuscript received: September 16, 2023; Revised January 3, 2024; Accepted February 12, 2024.

This paper was recommended for publication by Editor Jaydev P. Desai upon evaluation of the Associate Editor and Reviewers' comments.

[1]The authors are with Robotic Systems Lab, ETH Zurich, 8092 Zurich, Switzerland (e-mail: fannan@ethz.ch; mahutter@ethz.ch)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101070405. This research was also supported by the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab).

Digital Object Identifier (DOI): see top of this page.

A. Contributions

In this work, we propose a learning-based adaptive controller for hydraulic cylinders on heavy machinery that is trained purely in simulation and can adapt to real machines rapidly when deployed. Using a first-principle-based model of hydraulic cylinders, we simulate a large variety of nonlinear valve and cylinder dynamics. Then, Reinforcement Learning (RL) is used to train a controller that handles the different cylinder dynamics conditioned on a latent variable that encodes the system dynamics. During online deployment, we use an energy-function-based adaptation module to update the estimated latent parameter iteratively. We use the proposed method to train one universal controller and evaluate its performance in both simulation and real-world experiments on two machines that are very different in size, application, and age. When combined with a simple kinematic controller, the proposed method shows accuracy in end-effector tracking tasks comparable to existing methods that take machine-specific data for offline training. As the first demonstration of a generalizable controller for hydraulic cylinders, we believe this work opens up new possibilities for effortless and ubiquitous automation of hydraulic machinery.

II. Related Work

A. Control of hydraulic machinery

A number of works have been conducted in the direction of controlling hydraulic machines. The most commonly studied task is the control of the excavator arms. Early works in this direction focused on the kinodynamic control of the cranes [10], [11]. The model of actuators was assumed to be known, allowing the design of model-based controllers. However, the physical parameters are often not available in practice and are subject to change in reality.

A practical approach used in various applications is manually tuning a controller with feedforward and feedback blocks [5], [4]. In [4], a feedforward control signal is generated using a manually recorded look-up table, while a PID controller compensates for the remaining error. This approach was successfully used with an Inverse Kinematics (IK) controller in surface finishing tasks, achieving centimeter-level error. This approach is also used by some commercially available excavator automation solutions, such as the Leica iCon site excavator system [12]. The main limitation of this approach is the effort required to manually collect the feedforward controller and tune the feedback controller for each machine. It is shown in [13] that the feedforward can be generated automatially, if the other parameters in the actuator dynamics are available.

In recent years, machine learning approaches for modeling hydraulic dynamics have received a lot of attention. Egli and Hutter [6] proposed a data-driven modeling approach for hydraulic systems using Neural Networks (NNs). The NN model of the machine dynamics is then used in a simulation environment to train a learned controller with RL. Similarly, [7], [8] used hybrid model structures to reduce the number of trainable parameters and achieve faster training. The learning-based methods achieve precise control of the excavators, but they are still limited by the requirement to collect a machine-specific dataset.

B. Domain adaptation for learning-based control

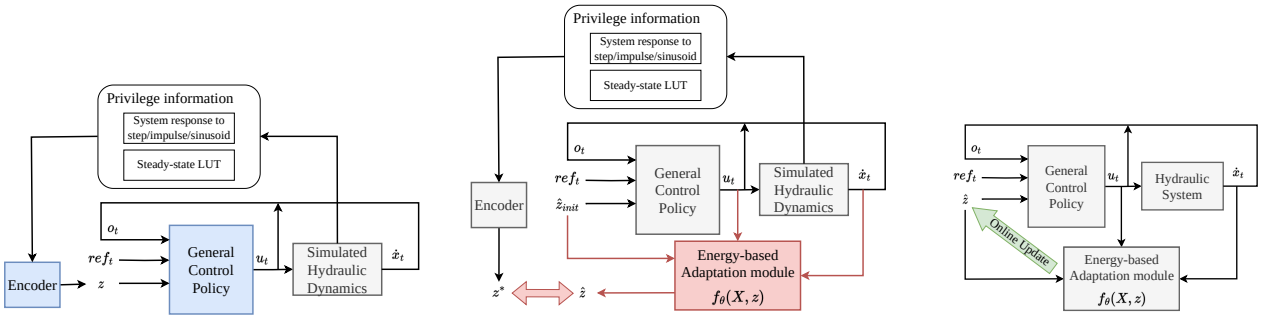
Adaptive control methods can deal with systems with time-varying or initially unknown parameters. Model-based adaptive control has been extensively studied and successfully applied to real-world control systems [14], [15]. There have been a few works that apply similar ideas in learning-based control. Yu et al. proposed the use of a universal control policy dependent on parameters related to environment or system dynamics. Online estimation over these parameters can be then performed with NN prediction [16] or Bayesian optimization [17], [18]. Guo et al. further showed the potential to use a differentiable simulator to identify system parameters that can be used by a universal policy [19]. Such approaches are able to adapt to unknown system parameters, but they have been primarily tested in simulation and with few unknown parameters. In reality, the unknown parameter space of the system is often more complex and has higher dimensions, making Bayesian optimization slow. In a system with high nonlinearities and heavy delay, such as hydraulic systems, the unknown parameters are also difficult to predict from a small amount of observed data.

III. Methodology

Fig. 2 shows an overview of the proposed method. We control hydraulic cylinders with an NN policy that tracks cylinder velocity commands. During training, we simulate the dynamics based on the first-principle model of the hydraulic actuators and heavily randomize the parameters. In addition to system state and reference velocities, the control policy also takes as input a latent variable z encoding the system dynamics. After obtaining a control policy, we train an adaptation module to estimate the latent variable using data that can be measured online. The adaptation module uses an energy-based model [20] that takes recent observations of the system and a current estimation of the latent variable as input, and it is trained to predict lower energy value for a latent variable that matches the observations. During online adaptation on real systems, the gradient of the energy function is used to update the latent estimation online.

A. Simulation Environment

For modern hydraulic machinery with load sensing systems and electrical pilot stages, a reduced-order model can be used to approximate the dynamics. Fig. 3 shows the diagram of a hydraulic cylinder on heavy machinery. The input to the hydraulic cylinder u is the current command to the solenoid valve that controls the pilot stage. The pilot stage moves the valve of the main stage, creating an orifice of size d, allowing oil flow between the pressure source and the piston. Following the simplifications in [21], we assume a closed hydraulic circuit, i.e. $Q_L = Q_A = Q_B$, and use the average effective piston area $\bar{A}$ and total hydraulic cylinder volume V_t. The



(a) Training control policy (b) Training adaptation module (c) Online deployment

Fig. 2. Overview of different phases in the proposed method. (a) In the first part of training, we simulate randomized hydraulic cylinder dynamics and train a control policy together that conditions on a latent variable that encodes the system dynamics. (b) In the second part of training, we train an energy-based adaptation module to estimate the latent variable using data available online. (c) To adapt the controller on a real system, the adaptation module is used to update the latent variable.

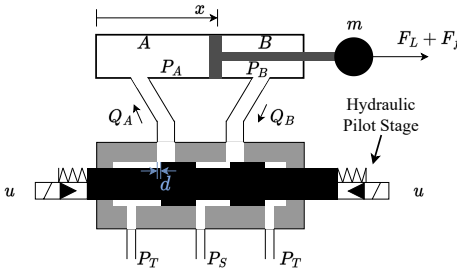


Fig. 3. Diagram of a generic hydraulic cylinder used on heavy machinery. Figure adapted from [21].

valve and piston dynamics can be described by the following ODEs:

$$Q_L = dK\sqrt{P_S - P_T - sign(d)P_L} \tag{1}$$

$$\dot{P}_L = \frac{4E}{V_t}\left(Q_L - \bar{A}\dot{x}\right) \tag{2}$$

$$\ddot{x} = \frac{1}{m}(\bar{A}P_L + F_L + F_f), \tag{3}$$

where $P_L = P_A - P_B$ is the load pressure, and F_f the friction force. m models the effective mass of the piston and attached mechanical structure, of which the inertias of the rotating parts are projected as masses on the cylinder. F_L consists of all external force projected to the piston. P_S and P_T are the source and tank pressure, respectively. The source pressure is regulated by the load sensing system based on load pressure measurements, such that $\sqrt{P_S - P_T - sign(d)P_L}$ remains constant. In this case, the reduced-order dynamics from d to $\dot{x}$ is described by the following second-order system:

$$m\dddot{x} = \frac{4E\bar{A}}{V_t}K'd - \frac{4E\bar{A}^2}{V_t}\dot{x} + \dot{F}_L + \dot{F}_f, \tag{4}$$

in which K' is the effective valve gain under load sensing.

On the tested machines in this work, the varying effective masses on the cylinders cause less than 30% change on system (4)'s eigenfrequency from the value at the nominal configuration. Therefore, we ignore the influence of varying effective mass m between different machine configurations and consider m constant for each actuator, although this assumption does not hold if a payload with large inertia is considered. The ODE in (4) also shows that only the time derivative of the external force F_L and friction force F_f affect the dynamics when starting from a static state. As we do not expect hydraulic machines to perform rapid motion while requiring high accuracy, we assume the velocity and acceleration of the cylinders are both low. Therefore, F_L caused by the gravity of the mechanical structure attached to the cylinder only slowly varies over time. With randomized disturbance added to the input during simulation, we expect the controller to be robust to these changes under slow operation. We consider a simple friction model $F_f = -\mu\dot{x} - F_s sign(\dot{x})$, where μ is the damping coefficient and F_s the static friction. While a more complex friction model could be more accurate, we found the simple model sufficient as the most significant influence of friction occurs only when the static friction abruptly changes direction.

The pilot stage dynamics have a similar form. However, as the pilot stage only drives the main stage valve, the oil flow is much smaller. Therefore, we assume that the pilot stage dynamics are much faster than the main stage dynamics and simulate the dynamics from u to d with a time delay and a dead zone. Finally, we add nonlinear mappings into the dynamics to simulate the nonlinearities caused by the valve in real-world systems. The nonlinear mappings are represented by piecewise linear functions with random positive slopes and added before and after the ODE system (4), similar to [7]. The monotonicity assumption is valid as the steady-state response should be monotonically increasing. The full model we used to train the controller in simulation is shown in Fig. 4.

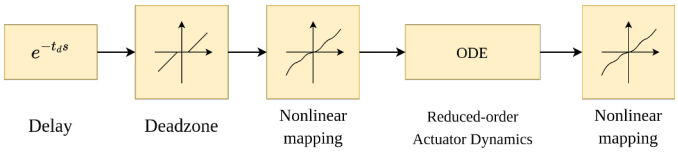


Fig. 4. Structure of simulated system model.

The parameters in the simulated model are randomized to represent various real-world systems. Specifically, we randomize the time of input delay, positive and negative dead zones, the slopes of the piecewise linear mappings, parameters in the linear system, as well as friction parameters. Based on prior experience with the autonomous excavator HEAP [4], we select the range of input delay up to $0.6\,\mathrm{s}$, and dead zones

up to 25% of the full input range. Later, we show that the developed controller is generalizable by testing it on another machine.

B. Control Policy

The control policy is trained in simulation to control various hydraulic actuators conditioning on a latent encoding of the system dynamics. We achieved this by training a latent variable encoder at the same time. The control policy, represented by an MLP with hidden layer dimensions $[64, 32]$, takes an input consisting of three parts: o_t, ref_t, and z. $o_t = [\dot{x}_t, u_{t-1}, \dot{x}_{t-1}, u_{t-2}, \ldots, \dot{x}_{t-k+1}, u_{t-k}]$ is a short history of the system's input-output observations, and $ref_t = [\dot{x}^*_t, \dot{x}^*_{t+1}, \ldots, \dot{x}^*_{t+N}]$ is a lookahead of the desired cylinder velocity. The memory and lookahead are necessary due to the randomized time delay in the system, and parameters k and N are selected to cover the maximum delay time we sampled in the simulation environment. The latent variable z used by the control policy is computed once for each random hydraulic model. After sampling the parameter set for each environment, we compute the steady-state response of the system to a set of constant inputs $u = -1.0, -0.9, \ldots, 0.9, 1.0$ and concatenate them with the transient response of the system to step and sinusoidal inputs at varying amplitudes and frequencies. This information is fed through the latent variable encoder to obtain a 16-dimensional latent variable z. The latent variable encoder first processes the transient response signals with a small Convolutional Neural Network (CNN). The output of the CNN is then concatenated with the steady-state response and fed through an MLP with hidden layer dimensions $[96, 64]$.

During training, random velocity references are generated. The reference velocity consists of constant velocity commands and ramps with constant accelerations. The acceleration is resampled every second from a uniform distribution with maximum and minimum values scaled by the relative gains of the simulated system. The reward function for the training is defined as:

$$r_k = r_{\dot{x}} + r_{\ddot{x}} + r_u, \tag{5}$$

where

$$r_{\dot{x}} = 0.5 \exp\left(-40||\dot{x}_k - \dot{x}^*_k||^2\right), \tag{6}$$

$$r_{\ddot{x}} = 0.2 \exp\left(-5||\dot{x}_k - \dot{x}_{k-1} - \dot{x}^*_k + \dot{x}^*_{k-1}||^2\right), \tag{7}$$

$$r_u = 0.1 \exp\left(-2||u_k - u_{k-1}||^2\right). \tag{8}$$

$\dot{x}_k$ is the cylinder velocity at time step k, and $\dot{x}^*_k$ is the reference velocity at time step k. The three terms in the reward function encourage velocity tracking, acceleration tracking, and smoothness of the control input, respectively.

We train the control policy and the latent variable encoder as a whole with RL. The Proximal Policy Optimization (PPO) [22] algorithm with Generalized Advantage Estimation (GAE) [23] is used[1]. Parallelized simulation was used to speed up the training and collect data from diverse environments. The training takes $0.5\,\mathrm{h}$ on a performant PC[2].

[1]The implementation we used could be found at https://github.com/leggedrobotics/RSLGym.

[2]The time is recorded when training on a desktop PC with Intel Core i9-12900KF CPU and an NVidia RTX 4090 GPU.

C. Adaptation Module

The privileged information used by the latent variable encoder requires significant efforts to obtain on a real machine. A series of experiments with different control inputs have to be conducted, and some of them are not always possible due to the physical limits of a hydraulic machine. Therefore, we use an energy-based adaptation module to estimate the latent variable at runtime using online-collected data. We have selected the energy-based modeling approach to estimate the latent variable because the latent variable should encode the dynamics of actuators, which is time-invariant. Therefore, during online deployment, the latent can be continuously updated until a minimum is found.

The adaptation module is a neural network that takes as input a concatenated vector of $[o_t, u_t \dot{x}_{t+1}, z]$ and predicts an energy value $f_\theta(o_t, u_t, \dot{x}_{t+1}, z)$, where θ is the network parameter. The energy function is trained to be minimized when the latent variable is close to that of the simulated system, such that the latent variable can be updated online by gradient descent on the energy function. To train the energy function, we use Algorithm 1.

Algorithm 1 Adaptation module training

for $i \leftarrow 1, \ldots, N_{iterations}$ **do**
  Sample N_{envs} systems
  Compute $z^*_1, \ldots, z^*_{N_{envs}}$ ▷ With trained encoder
  Collect $o^j_t, u^j_t, \dot{x}^j_{t+1}$
    $\hookrightarrow$ for $j = 1, \ldots, N_{envs}$, $t = 1, \ldots, T$
  Sample $z^0_1, \ldots, z^0_{N_{envs}}$
  for $j \leftarrow 1, \ldots, N_{steps}$ **do**
    for $k \leftarrow 1, \ldots, N_{envs}$ **do**
      $z^j_k \leftarrow z^{j-1}_k - \gamma \nabla_z \left[\sum_t f_\theta(o_t, u_t, \dot{x}_{t+1}, z^{j-1}_k)\right]$
    end for
  end for
  $\theta \leftarrow \theta - \alpha \nabla_\theta \left[\sum_k ||z^*_k - z^{N_{steps}}_k||^2\right]$
end for

During each parameter update of the function f_θ, we sample N_{envs} systems from the simulation environment and compute z^* for each of them using the trained latent encoder. Then, we collect rollouts of T steps on each system. With the collected data, N_{steps} steps of gradient descent on the latent variable are performed to minimize the energy function, starting from a random guess. The loss is computed as the distance between the updated latent estimation to the ground truth z^* and used to update θ. We structurally constrained the energy function to be convex with respect to z using the method proposed in [24]. We also found this constraint helpful in making the energy function conservative when fitting the multi-modal data. As the data we fed into the energy function is from a short period of simulation, there could exist multiple local minima in the z space that correspond to the same observed data. In these cases, constraining the energy function to be convex forces it to be conservative when the data is not sufficient to distinguish the system dynamics.

We selected hyperparameters $N_{env} = 3000$, $T = 200$, $N_{steps} = 5$, $\gamma = 1.5e{-}4$, $\alpha = 2e{-}3$ and $N_{iterations} = 2400$.

The training takes 3.5 h on a PC. During online deployment, we perform the gradient evaluation on a short history of data every 1 s, i.e.

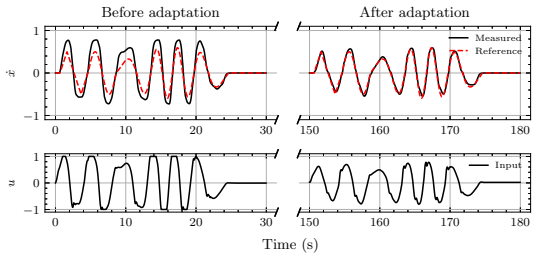
$$\hat{z}_t \leftarrow \hat{z}_{t-1} - \gamma \nabla_z \left(\Sigma_{\tau \in [t-16,t]} f_\theta(o_\tau, u_\tau, \dot{x}_{\tau+t_s}, \hat{z}_{t-1}) \right) \quad (9)$$

IV. Simulation Result

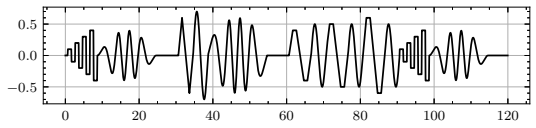
The performance of the proposed method is first evaluated in simulation. We use both the simplified model used in training and an NN model trained from extensive experimental data. During the evaluations, we initialize the latent variable estimation at its empirical mean from the randomized training stages. While the controller predicts control commands at every time step, a separate thread records the data and periodically updates the latent variable estimation. In the simulation, we use a combination of steps, ramp signals, and modulated sinusoids as the reference signal. Varying magnitude and frequencies of the reference signal are used during the adaptation.

A. Simulated System Model

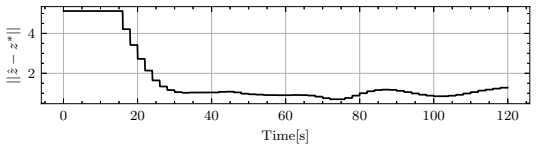
Fig. 5 shows the performance of the proposed controller in an example of the simulated systems. Fig. 5(a) shows that in the beginning, the tracking performance is poor because the system dynamics are randomly sampled. During the 2-minute adaptation, a reference signal consisting of steps, ramps, and sinusoids with varying magnitude and frequencies, as shown in Fig. 5(b), is commanded to the system. With the observed data, the latent variable estimation gets closer to the ground truth as shown in Fig. 5(c). Finally, the tracking performance has improved significantly in the right half of Fig. 5(a).



(a) Reference velocity tracking performance on a simulated system, before and after adaptation. The reference and tracked velocities are shown in the first row, and the input is shown in the second.



(b) Reference signal used during the adaptation process.



(c) Convergence of latent variable during the adaptation, shown by the 2-norm of the difference between estimated latent and the ground truth predicted by the encoder.

Fig. 5. Performance of proposed adaptive control method on a simulated system.

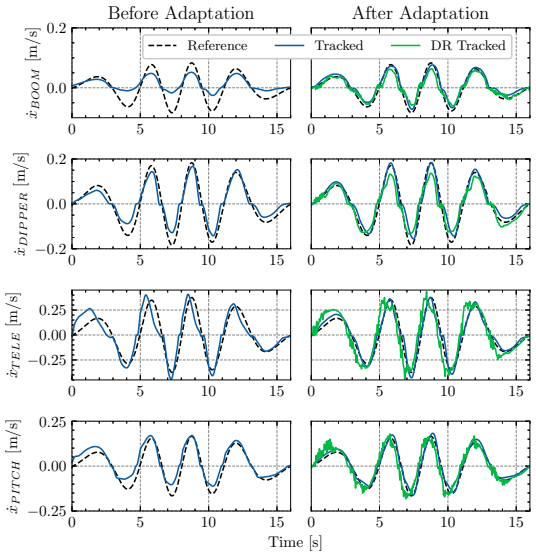


Fig. 6. Adaptation performance on actuator network of HEAP. Each subplot shows the tracking performance of the proposed controller on one of the four boom cylinders. The left half of each subplot shows the performance before adaptation, and the right half shows the performance after adaptation and the performance of the controller trained with DR.

B. Actuator Network

We tested the proposed method on the actuator network taken from [6], a system resembling an actual hydraulic machine. The actuator network is trained to predict the cylinder dynamics of the four boom cylinders on HEAP. The model is a 4-layer MLP and captures complex effects on a real hydraulic machine, and controllers tested on this model have been successfully transferred to the machine.

We perform the same test as section IV-A, and Fig. 6 shows the tracking performance of the proposed controller on the actuator network before and after adaptation. Similar to the previous result, the proposed controller can adapt to the system and largely improve the tracking performance. This result suggests that the proposed model in III-A is a good abstraction of more complex hydraulic dynamics, and the adaptive controller trained on this model can be applied to real hydraulic machines.

We also compare the performance of the adaptive controller with another controller trained with Domain Randomization (DR). The controller trained with the same randomized environment in Section IV, but the policy does not depend on the latent variable. The cylinder velocity tracking performance is plotted in the right half of Fig. 6 for a comparison with the adapted controller. Compared to the controller trained with DR, the proposed controller has a lower RMS tracking error on all four cylinders. The level of improvement is from 17.5% on the boom cylinder to 59.7% on the pitch cylinder. As the DR controller does not have implicit knowledge of the actuator dynamics, it cannot generate a good feedforward input for the system and rely heavily on feedback control. This leads to the temporal offset, overshooting, and oscillation

in reference tracking, particularly in the telescopic and pitch cylinders. These effects lead to bigger oscillations on the end-effector when all cylinders are used coordinately. Thus, it is not possible to use the DR controller for precise motion control.

V. Experiment Result

We perform the real-world experiments on two different machines, the autonomous excavator HEAP [4], and an automated tree harvester SAHA (**S**upervised **A**utonomous **HA**rvester) [25]. On each machine, we first perform the adaptation on the cylinders and then test the adapted controller in end-effector trajectory following tasks resembling real-world operations.

During the adaptation phase, each joint is commanded to follow ramp signals and modulated sinusoids with different frequencies and amplitudes. This process is supervised by a human operator who adjusts the references with safety considerations of the machine kinematics. The adaptation is stopped once the control performance stops improving and the latent estimation converges. The adaptation is performed on each joint individually also for safety reasons.

In the end-effector control experiments, each joint is controlled independently by one instance of the proposed controller with the adapted latent variable. The user commands a reference trajectory, and IK is used to map the reference trajectories to cylinder space. The velocity reference of each cylinder controller consists of a feedforward part from the trajectory and a feedback command from a PID controller tracking cylinder position. To guarantee the continuity of the reference velocity, the reference trajectory always consists of an acceleration phase to a constant reference velocity at the beginning and a constant deceleration phase at the end.

A. HEAP

On the excavator HEAP, the proposed cylinder velocity controller adapts to the dynamics of the arm actuators on the real machine within 10 minutes. The adapted cylinder controller performance is shown in Fig. 7. It can be observed that the cylinder controller can track the reference velocity well despite small mismatches at the peaks of the reference velocities. The friction on the real machine is also significant, indicated by the sudden accelerations when the cylinder velocity crosses zero.

The cylinder controller also performs decently well when all the cylinders are used simultaneously. Fig. 8 shows an experiment in which a circular reference trajectory is tracked with the shovel edge of HEAP using the proposed method. With the high performance of the cylinder controller, the mean end-effector trajectory tracking error over the trajectory shown in Fig. 8(a) reaches $4.1\,\mathrm{cm}$, which is comparable to the previous work on the same machine [6].

A series of horizontal straight-line tracking experiments at different heights and reference velocities were performed, and we computed the same performance metrics as used in [6] for a comparison. Some of the tracked trajectories on the machine are plotted in Fig. 9. The error is computed and listed in Table I, together with the values from previous work. In Table

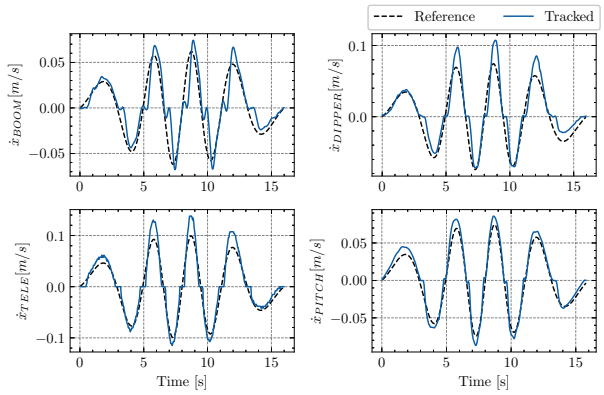
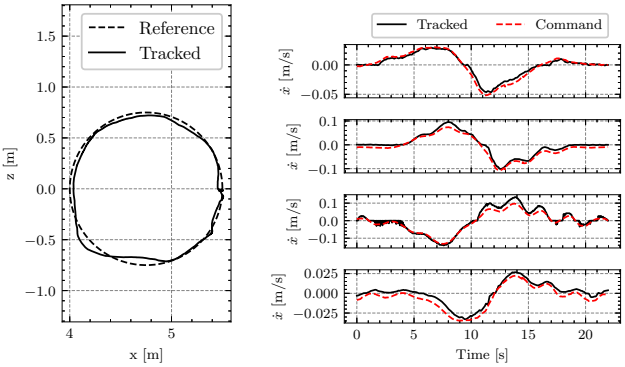


Fig. 7. Cylinder velocity tracking performance on HEAP after adaptation. Each subplot shows one active cylinder on the arm of HEAP. The reference velocity is a modulated sinusoid.



(a) Trajectory tracking (b) Cylinder velocity tracking

Fig. 8. Circular trajectory tracking performance of the proposed controller. (a) The measured shovel edge trajectory and the reference trajectory. (b) Measured cylinder velocities and the reference cylinder velocities. Each subplot in (b) shows one active cylinder on the arm of HEAP. The reference trajectory is a circle with a diameter of $1.5\,\mathrm{m}$. The reference velocity is $40\,\mathrm{cm\,s^{-1}}$ and the reference acceleration is $5\,\mathrm{cm/s^2}$. The trajectory is defined in the cabin frame, where x axis is defined as the horizontal direction of the excavator arm extension, and z is the vertical direction pointing up.

I, v is the velocity of the reference trajectory, e_p the path deviation, and $\dot{\theta}$ the end-effector angular rate. Despite some differences in the experimental setup, the data suggest that the proposed controller can reach tracking accuracy comparable to that of the benchmarked methods. The mean path deviation of the proposed method outperforms the commercial solution and even outperforms the controller trained on the actuator network at a slow speed. The shovel angle tracking using the proposed method is also clearly better than the benchmarked methods. The maximum path deviation using the proposed method is higher than that using the controller trained on the actuator network, but it is still at a similar level as the accuracy of the commercial solution. Fig. 9 suggests it mostly comes from the larger position error at the beginning of the trajectory. This can be explained by the fact that we control each cylinder independently in the experiment, and it can be further improved with a high-level controller that uses all actuators coordinately.

The method also works in the presence of ground interaction. Fig. 10 shows a grading experiment on HEAP, where the reference trajectory is a straight line around $5\,\mathrm{cm}$ beneath an uneven ground surface. An average trajectory tracking error

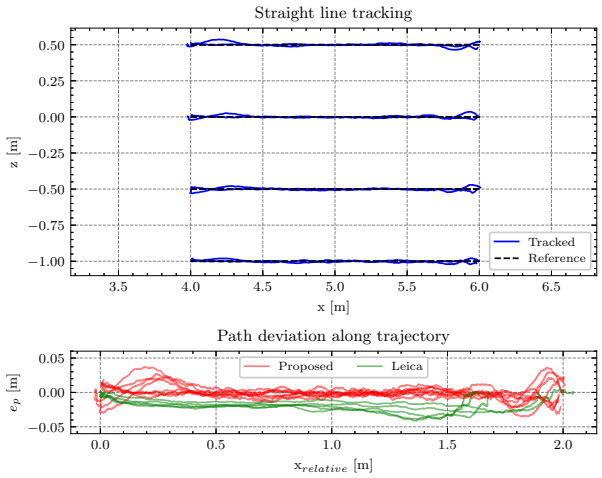


Fig. 9. Result of straight line tracking experiment with the proposed controller. (Top) The reference trajectories are straight lines with a length of 2 m and with reference velocity at $15\,\mathrm{cm\,s^{-1}}$ with $7.5\,\mathrm{cm/s^2}$ reference acceleration. Each trajectory is tracked in both directions. The plot corresponds to the 'Slow' case in Table I. (Bottom) Path deviation along the tracked trajectory, and comparison with the Leica iCon system. The trajectories are not the same length as the Leica system is semi-autonomous.

TABLE I
ERROR CALCULATION FOR THE STRAIGHT LINE TRACKING EXPERIMENT AND COMPARISON WITH LEICA ICON SYSTEM. THE DATA OF THE BENCHMARK METHODS IS TAKEN FROM [6].

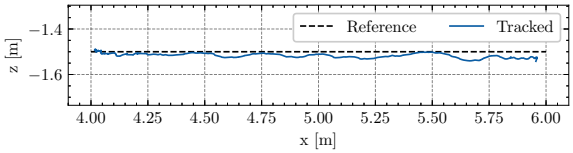
Metric	Proposed		Egli & Hutter[6]		Leica
	Slow	Fast	Slow	Fast	Fast
v^{avg} [cm/s]	13.04	20.00	13.1	18.3	16.3
v^{max} [cm/s]	15.00	25.00	24.9	35.1	40.9
$\lvert e_p\rvert^{avg}$ [cm]	0.69	1.16	0.8	0.9	1.8
$\lvert e_p\rvert^{max}$ [cm]	3.73	5.91	1.7	2.2	4.2
$\lvert\dot{\theta}\rvert^{avg}$[rad/s]	0.004	0.007	0.010	0.012	0.015
$\lvert\dot{\theta}\rvert^{max}$[rad/s]	0.033	0.046	0.030	0.041	0.046

at 3.62 cm was reached, while the average deviation from the straight line path is 3.38 cm. This level of accuracy is sufficient for practical grading tasks. The proposed method, however, does not work for excavation tasks or grading with higher ground penetration. In these tasks, the ground interaction force grows rapidly. This effect dominates the dynamics in (4), and the assumption on slowly varying F_L is no longer valid. In addition, the rapidly changing load force also causes the load sensing system not to act perfectly, which further deteriorates the performance of the proposed method.

B. SAHA

SAHA, as shown in Fig. 11, is a 4.5-ton small forest machine retrofitted recently with sensors and control interfaces for robotic precision forestry. To validate the capability of the proposed controller to generalize on different machines, similar experiments were repeated on SAHA. We used the exact same trained controller that has been successfully deployed on HEAP. As forest machines are equipped with passively suspended end-effectors, we only consider the arm up to the last controllable joint when performing task-space control.

(a) Photo of the grading experiment on HEAP



(b) Grading trajectory

Fig. 10. Photo of a grading experiment on HEAP and the recorded trajectory. The reference trajectory is a straight line with a length of 2 m and a ground penetration around 5 cm. The reference velocity is $20\,\mathrm{cm\,s^{-1}}$.

Fig. 11. A picture of SAHA with the controllable joints marked in yellow. The red circle shows the mounting point of the underactuated cutting tool. In the experiments, we control the active cylinders on the arm to track task space trajectories (red dashed line) with the tool mounting point.

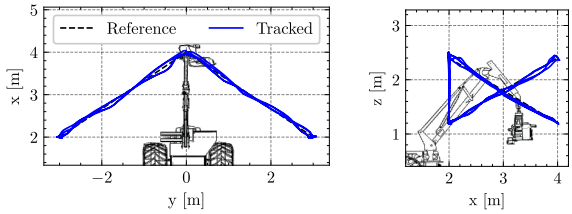


Fig. 12. Task space tracking performance of the proposed controller on SAHA. The reference trajectory is a set of line segments defined by a series of waypoints. The trajectory is repeated three times. The two subplots show the top view and the side view of the trajectory tracking experiment. The coordinate frame is defined such that x points at the forward direction of the chassis, and z points upward. The visualization of the SAHA machine is for illustration only, and the scale is not accurate.

We performed the same adaptation process on the SAHA machine, which took about 8 minutes until the cylinder controller could effectively track the reference velocities. After that, we performed end-effector trajectory tracking experiments on the arm of SAHA. We selected a trajectory that involves line segments through a set of waypoints in front of the chassis, initiating the action of reaching out to different trees in a forestry operation. Fig. 12 shows the performance with $0.6\,\mathrm{m\,s^{-1}}$ reference velocity and $0.3\,\mathrm{cm/s^2}$ acceleration. The average position tracking error is 5.39 cm.

The higher tracking error on SAHA is unavoidable as the

machine is older with outdated hydraulic components and significant backlash. The passively suspended end-effector also introduces a changing load force to the system, which potentially leads to a higher tracking error. Despite the less accurate trajectory tracking, the controller still steers the arm to the waypoints accurately. The result on SAHA shows the excellent generalizability of the controller trained with the proposed method. Thanks to the adaptation process, the same controller can be conveniently deployed on machines with very different actuator designs for different tasks.

VI. Conclusion

In this work, we present a method to train adaptive learning-based controllers for the automation of hydraulic machinery. Through training in a randomized simulation with simplified hydraulic dynamics, the resulting controller can be adapted to different real systems within minutes. The accuracy of the adapted controller is comparable with the controller trained with a machine-specific actuator model and is adequate for many applications. Through the test on multiple machines, we show the excellent capability of the proposed method to automate multiple machines with the same controller. We believe the proposed method has great value in the automation of hydraulic machinery, as machine operators can easily adapt the controller to their systems without expert knowledge.

The proposed method can still be improved in several directions. Pressure sensors on hydraulic machines provide additional observation of the system state at a reasonable cost. As the actuator dynamics can be dependent on the load conditions pump output, the additional measurement of cylinder and pump pressures can be used to improve the model in this work. This would enable the use of the adaptive controller to perform more diverse tasks including excavation, as done in [7] for a single machine. The proposed pipeline could be applied to continuously refine the performance of the control policy with data collected during operation, such that slow variations in the machine dynamics due to aging, temperature changes, etc., can be compensated seamlessly. For this, long-term evaluations need to be carried out, and the method should be improved to handle data from real operations, which can be biased since it does not excite the actuators at different modes. Finally, the adaptation capability of the current method is still limited by randomization during training. To overcome this limitation, we consider online learning a promising approach to building controllers that can adapt to a broader range of systems.

ACKNOWLEDGMENT

The authors would like to thank Pascal Egli, Pol Eyschen, Hao Ma, and Michael Muehlebach for the discussions during the project, and the support with the hardware experiments.

References

[1] P. Digvijay. (2021) Construction Robotics Market. (accessed 2023-04-20). [Online]. Available: https://www.alliedmarketresearch.com/construction-robotics-market-A09408

[2] R. Parker, K. Bayne, and P. W. Clinton, "Robotics in forestry," *New Zealand Journal of Forestry*, vol. 60, no. 4, pp. 8–14, 2016.

[3] M. Bergerman, J. Billingsley, J. Reid, and E. van Henten, "Robotics in Agriculture and Forestry," in *Springer Handbook of Robotics*, ser. Springer Handbooks, B. Siciliano and O. Khatib, Eds. Cham: Springer International Publishing, 2016, pp. 1463–1492.

[4] D. Jud, S. Kerscher, M. Wermelinger, E. Jelavic, P. Egli, P. Leemann, G. Hottiger, and M. Hutter, "HEAP - The autonomous walking excavator," *Automation in Construction*, vol. 129, p. 103783, 2021.

[5] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, and D. Manocha, "An autonomous excavator system for material loading tasks," *Science Robotics*, vol. 6, no. 55, p. eabc3164, 2021.

[6] P. Egli and M. Hutter, "A General Approach for the Automation of Hydraulic Excavator Arms Using Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5679–5686, 2022.

[7] M. Lee, H. Choi, C. Kim, J. Moon, D. Kim, and D. Lee, "Precision Motion Control of Robotized Industrial Hydraulic Excavators via Data-Driven Model Inversion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1912–1919, 2022.

[8] J. Weigand, J. Raible, N. Zantopp, O. Demir, A. Trachte, A. Wagner, and M. Ruskowski, "Hybrid Data-Driven Modelling for Inverse Control of Hydraulic Excavators," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2127–2134.

[9] A. Taheri, P. Gustafsson, M. Rösth, R. Ghabcheloo, and J. Pajarinen, "Nonlinear Model Learning for Compensation and Feedforward Control of Real-World Hydraulic Actuators Using Gaussian Processes," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9525–9532, 2022.

[10] Q. Ha, M. Santos, Q. Nguyen, D. Rye, and H. Durrant-Whyte, "Robotic excavation in construction automation," *IEEE Robotics & Automation Magazine*, vol. 9, no. 1, pp. 20–28, 2002.

[11] P. H. Chang and S.-J. Lee, "A straight-line motion tracking control of hydraulic excavator system," *Mechatronics*, vol. 12, no. 1, pp. 119–138, 2002.

[12] Leica iCON site excavator - 3D System. (accessed 2023-06-26). [Online]. Available: https://leica-geosystems.com/products/machine-control-systems/excavator/leica-icon-ixe3---3d-system

[13] J. Nurmi and J. Mattila, "Automated Feed-Forward Learning for Pressure-Compensated Mobile Hydraulic Valves With Significant Dead-Zone," in *ASME/BATH 2017 Symposium on Fluid Power and Motion Control*. American Society of Mechanical Engineers Digital Collection, 2017.

[14] K. J. Åström and B. Wittenmark, *Adaptive Control: Second Edition*. Courier Corporation, 2013.

[15] E. Lavretsky and K. A. Wise, *Robust and Adaptive Control: With Aerospace Applications*, ser. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2013.

[16] W. Yu, J. Tan, C. Karen Liu, and G. Turk, "Preparing for the Unknown: Learning a Universal Policy with Online System Identification," in *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, 2017.

[17] W. Yu, C. K. Liu, and G. Turk, "Policy Transfer with Strategy Optimization," in *International Conference on Learning Representations*, 2018.

[18] W. Yu, J. Tan, Y. Bai, E. Coumans, and S. Ha, "Learning Fast Adaptation With Meta Strategy Optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2950–2957, 2020.

[19] M. Guo, W. Yu, D. Ho, J. Wu, Y. Bai, K. Liu, and W. Lu, "Universal Controllers with Differentiable Physics for Online System Identification," 2021.

[20] Y. Du and I. Mordatch, "Implicit Generation and Modeling with Energy Based Models," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[21] M. Ruderman, "Full- and reduced-order model of hydraulic cylinder for motion control," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 7275–7280.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017.

[23] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," *arXiv:1506.02438*, 2018.

[24] B. Amos, L. Xu, and J. Z. Kolter, "Input Convex Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017, pp. 146–155.

[25] E. Jelavic, T. Kapgen, S. Kerscher, D. Jud, and M. Hutter, "Harveri : A Small (Semi-)Autonomous Precision Tree Harvester," in *Innovation in Forestry Robotics: Research and Industry Adoption, ICRA 2022 IFRRIA Workshop*, 2022.